DATA SCIENCE JOURNAL

Kunze, J et al 2017 Persistence Statements: Describing Digital Stickiness. *Data Science Journal,* 16: 39, pp. 1–11, DOI: https://doi.org/10.5334/dsj-2017-039

ESSAY

Persistence Statements: Describing Digital Stickiness

John Kunze[1], Scout Calvert[2], Jeremy D. DeBarry[3], Matthew Hanlon[4], Greg Janée[1] and Sandra Sweat[5]

[1] University of California Office of the President, US

[2] Michigan State University, US

[3] University of Georgia Athens, GA, US

[4] Texas Advanced Computing Center, Austin, TX, US

[5] Athena Health, US

Corresponding author: John Kunze (jak@ucop.edu)

In this paper we present a draft vocabulary for making "persistence statements." These are simple tools for pragmatically addressing the concern that anyone feels upon experiencing a broken web link. Scholars increasingly use scientific and cultural assets in digital form, but choosing which among many objects to cite for the long term can be difficult. There are few well-defined terms to describe the various kinds and qualities of persistence that object repositories and identifier resolvers do or don't provide. Given an object's identifier, one should be able to query a provider to retrieve human- and machine-readable information to help judge the level of service to expect and help gauge whether the identifier is durable enough, as a sort of long-term bet, to include in a citation. The vocabulary should enable providers to articulate persistence policies and set user expectations.

Keywords: persistent identifiers; metadata; standardization; content variance; versioning; commitment; linking; citation; referencing

Introduction

While persistence of cited objects is typically desirable (Ball 2010), it is not clear what persistence really means. It is not in and of itself helpful for an object to have a so-called persistent identifier (PID). Of course the object must be extant when we next need it, but its persistence is purely a matter of service, not conferred by or inherent in an identifier (Kunze 2002). For want of a better term, we will refer to "persistent identifiers" for objects that can change or disappear, irrespective of the intentions or purposes of the provider. Even scholarly articles suffer from link rot (Klein et al. 2014) and context drift (Jones et al. 2016) at astounding rates. But PIDs are the central approach taken to insulate users from the changes to which objects are subject, and we will be careful to distinguish between PIDs and the objects they identify.

Persistence as a service is a broad concept. At a minimum it implies a prediction about an archive's commitment and capacity to provide some specific kind of long-term functionality. A single provider might support long-term access to objects having a diversity of policies, in which some objects are strictly unchanging, others subject to correction, and a third class subject to significant update within a constant theme (e.g., permanent access to current weather conditions or to a home page). Across providers the range of policy and practice widens. Meanwhile providers lack a common standard terminology to distinguish one particular practice from another.

Different kinds of persistence

The concept of persistence is nuanced in ways often overlooked. Despite decades of debate (Moats 1997) about permanence of digital objects and identifiers, they are called "persistent" or "not persistent", as if persistence were a binary property, that is, either on or off. Nothing is permanent, yet one regularly hears naive assertions about object permanence being guaranteed by having a DOI (Bilder 2013), or identifier uniqueness being guaranteed by a UUID (universally unique identifier), all on top of digital infrastructure that no vendor warranties. So a thing's "permanence" or "persistence" is at best a suggestion that it will resist change. Perhaps we need new jargon, such as digital "stickiness" or "stubbornness". Of course, it's not things that resist change, it's providers of things that resist or, more precisely, control change. That digital resources change has long been a feature of the web architecture (Berners-Lee 1996). More recently, PIDs, their essential types of information, and Handle System resolution mechanisms have been the focus of recommendations from the Research Data Alliance's PID Information Types and Data Types Registries working groups (Weigel, DiLauro & Zastrow 2014; Broeder & Lannom 2014).

It is not just providers of objects who have diverse needs, but also scholars who cite them. On one hand the case has been made that reproducible science needs citations to datasets that do not change at all (Hey 2009). On the other hand, there is an emerging class of citations that essentially advertise datasets for which continual access and improvement is a feature; for example, "data papers" announce the availability of new data, databases, and synthesized datasets, and their citation is seen as way to get credit and attribution. Seldom discussed, but needed all the same, are long-term citations for general objects that returning visitors should expect to see improve, whether by maintenance mode corrections or by funded programs of enhancement (e.g., curated databases, software packages).

Some concrete examples of the range of strategies in use may be helpful. The DataONE federated data network (Michener et al. 2011) assigns a PID to immutable data objects and a "series identifier" that resolves to the latest version of an object (DataONE 2015). By contrast, repositories such as figshare (figshare 2016) and Merritt (Abrams et al. 2011) tolerate changes to metadata under the PID assigned originally, but create a new "versioned" PID if the object title or a component file changes, and in the latter case, the original non-versioned PID always references the latest version. On the other hand, the universal numeric fingerprint (Altman & King 2007) is a PID that supports citation of numeric data in a way that is largely immune to the syntactic formatting and packaging of the data. In the particular case of evolving narrative, the W3C Manual of Style stipulates (W3C 2014) authoring conventions that make it clear how to recover different document versions.

The need for a persistence vocabulary

These strategies are not necessarily visible or intelligible to users. Moreover, in current practice, between guessing at a provider's ability, divining its actual intentions, and predicting unforeseeable events, the citation of digital objects has become a near-futile exercise in forecasting. A key missing tool is terminology to describe the various kinds of persistence in a way that can help users decide which objects they will want to cite for the long-term. This paper describes a concise set of metadata terms and controlled values to express projected digital "stickiness" via "persistence statements". We propose this set of terms as part of a strategy for conveying accurate information as to the intentions of data providers (as with any statement, whether it is credible is mostly outside the scope of this paper).

A challenge is to define new terms that work as a group to cover common use cases, and that trade off precise specialized meaning against jargon that is natural enough but not too overloaded. One approach is to invent a term (e.g., a portmanteau word) or choose an existing word that is unusual enough (e.g., rare, archaic) to make the reader hesitate to jump to a conclusion about its meaning. If done successfully, this would create precise and distinct new jargon that feels somewhat natural and avoids overloading acronyms common in technological fields.

The term definitions under development have been made available for comment and refinement via the crowd-sourced YAMZ (yamz.net) metadata dictionary. Some of this draws on prior work (Byrnes 2000). The notion of a persistence statement queryable by an ARK identifier (Kunze 2002) is borrowed to apply here with any object identifier. In this instance "queryable" means that the identifier itself, with a specific kind of URL query string appended, can be submitted to a web browser.

With a means to convey more information about a provider's intentions and abilities to persistently provide access to a data object, an advantage is conferred over blanket terms of services (ToS) that bury a vague promise of "best effort" and nothing more. A way to convey dimensions of commitment and capability lets providers offer explicit information that helps users adjust their expectations.

In what follows, we first contextualize a means for querying object identifiers to return persistence information. We then describe how our vocabulary can help set user expectations along several dimensions, including content variance, object availability, and the growth patterns of digital objects. We have attempted to coalesce meanings for persistence around words that are unlikely to be mistaken for other commonly used terms in our field. We then consider tensions in policies on identifier assignment for changing content, and offer strategies for conveying version information to users or machines with terms in our vocabulary. These terms can be used to inflect a query that automatically returns a particular version of the object. Finally, we offer some strategies for a provider to relay information about itself, including succession planning and data certifications, how it prioritizes objects it provides, and how it assigns identifiers that can aid users in judging provider intentions and promises. As networked sources of digital objects proliferate, many with good intentions but limited ability to make long term promises in good faith, let alone seek certification of those promises, a means for conveying realistic commitments invites honest statements by providers and clear assessments by users and depositors.

Persistence queryable by object identifier

As mentioned, no object identifier string, regardless of scheme, can tell us much about the object's persistence (as forecasted by the provider). One promising scenario, however, is to use the identifier in a query that returns a "persistence statement" to help users judge whether and how to cite the object. For this we need some identifier notions.

- *id string*: the sequence of characters that is the identifier string itself, possibly modified by adding a well-known prefix (often starting with http://) in order to turn it into a URL.
- *identifier*: an association between an id string and a thing; e.g., an identifier "breaks" when the association breaks, but to act on an identifier requires its id string.
- *actionable identifier*: an identifier whose id string may be acted upon by widely available software systems such as web browsers; e.g., URLs are actionable identifiers.

If it were made queryable, the actionable id string would effectively lead to a story that conveys provider support policies, expected changes to the object (e.g., none, or corrections only), and the nature of the provider itself. This sort of sticky statement is not binary. Instead it is nuanced and dimensional, suggesting a breakdown into metadata elements.

We identify several kinds of machine- and human-readable metadata elements that would help users gauge the persistence commitments and abilities of repositories and archives. This includes support policies, the nature of the data provider, and, for a given object, the level of support intended and the kinds of change to expect. While element names, values, and precise semantics are in flux, they fall into several fixed categories. Most of the terms below are hyperlinked to lead to definitions published at YAMZ.net.

Setting user expectations: content variance

We define *content variance* to be a description of the ways in which provider policy or practice anticipates how an object's content will change over time. Approaches to content variance differ depending on the object, version, service, and provider.

To keep things simple, we assume an object's content (e.g., an article, dataset, or image) incorporates any provider-maintained, user-visible descriptive information; thus, adding a comma to an Author metadata element constitutes a change to the object, and whether that change warrants a new identifier or version is up to the provider. For a given identifier, the provider might assert one of the following policies:

- *frozen*: The bit stream representing previously recorded content will not change.
- *keeping*: Previously recorded content will not change, but character, compression, and markup encodings may change during a format migration, and high-priority security concerns will be acted upon (e.g., software virus decontamination, security patching).
- *fixing*: Previously recorded content may be corrected at any time, in addition to any change under "keeping".
- *rising*: Previously recorded content may be improved at any time, for example, with better metadata (datasets), new features (software), or new insights (pre- and post-prints). This encompasses any change under "fixing".
- *molting*: Previously recorded content may be entirely overwritten at any time with content that preserves thematic continuity. For example, an organization's homepage may be completely reworked while continuing to be its homepage, and a weather or financial service page may reflect dramatic changes in conditions several times a day.

Setting user expectations: object availability

Providers may (or may not) commit to keeping a given object available. They might even commit to removing an object, for example, by a certain date or upon first use. While this change could be seen as an extreme form of content variance, we prefer a separate descriptor for *object availability*, the period of time during which the provider expects to keep the object available.

- *finite*: availability is expected to end on or around a given date (e.g., limited support for software versions not marked "long term stable") or trigger event (e.g., single-use link).
- *indefinite*: the provider has no particular commitment to the object.
- *lifetime*: the object is expected to be available as long as the provider exists.
- *subinfinite*: due to succession arrangements, the object is expected to be available beyond the provider organization's lifetime.

Setting user expectations: objects that grow

An important dimension of content variance is growth. Constant growth is often seen as an extremely difficult problem in dynamic citation, but if a provider can declare that object growth that merely adds content to the end, the problem becomes tractable. We have a term for such growth:

- *waxing*: change that is limited to appending content in a way that does not in itself disrupt or displace previously recorded content. Examples of waxing objects include live sensor-based data feeds, citation databases, and serial publications.

Policies on object and version identifier assignment

There is a dualism between content variance and identifier assignment policy, which can be seen as opposing, interdependent forces. The further that content moves away (varies) from its original state the more likely the provider is to give it a new identifier or new version number (or version identifier). Conversely, the less that content drifts, the less pressure the provider will feel to assign a new identifier. Precisely when such assignment will be triggered depends on policy that will differ across objects, collections, and providers. Some of this thinking has origins in the ARK generic policy service (Kunze & Rodgers 2008).

Policies on and versioning of web content is not new. The "dated URI" (Masinter 2012) outlines a way to identify content that may not be web accessible. For web-accessible content versions, identified by timestamp, that happen to have been saved in one or more global archives (with or without the knowledge of the provider), the Memento framework is very helpful (Van de Sompel, Nelson & Sanderson 2013). This paper focuses on versions and policies assigned by providers. Regarding policy retrieval, a site could potentially register a "well-known" URL path (Nottingham & Hammer-Lahav 2010) for retrieving a persistence statement, but in our experience there is often not one uniform policy across all the objects, collections, and providers served by the site.

We position ourselves near the middle of a policy continuum. At one end of the continuum, any content change triggers generation of a new object identifier. There are no version numbers, all object content is frozen, and any changed content is viewed as creating a new object. This policy works well in fully automated settings (e.g., named-data.net), but many curated collections have policies towards the middle of the continuum. With curated database and software release collections, variance policy can describe kinds of expected change at more than one level; for example, with Ubuntu Linux software distributions, differences appropriate between *major* versions are more significant than differences tolerated between *minor* versions.

Towards the other end of the continuum, different content variance policies, expressible with our vocabulary, ought to apply separately to object identifiers, versions, and sub-versions. Moreover, the dualism suggests that content variance policy implies triggers for identifier and version assignment. If the provider permits defined change up to a certain threshold, once that threshold is passed for a stored object, a new identifier (or version) will have been assigned. Because of the complex multi-level nature of such assignment policy, we merely note this area without attempting to define terms for articulating it.

Referencing content in the presence of versions

While objects and versions are thus assigned identifiers according to a variety of policies, this need not be especially troubling for interoperation because of the importance of how content is referenced. Providers worry about digital objects, versions, and identifiers, but users are concerned with intellectual and artistic "content", a definition for which is overdue.

- *content*: abstract substance, found in such things as writing, speech, images, and music, as distinct from form or style.

For the purpose of discussion, cited (referenced) content that includes an actionable id string may be seen to fall into one of the following cases:

- *extraversioned*: a version identifier is separate from the id string, so that the actionable id does not lead to specific version without human intervention, e.g., "http://doi.org/10.2345/67", Version 4.
- *intraversioned*: a version identifier is part of the id string, e.g., "http://doi.org/10.2345/67.V4".
- *introversioned*: a kind of intraversioned content for which the version identifier (within the object identifier) is opaque, e.g., "http://doi.org/10.2345/678", which happens to be version 4.

These three kinds of practice (policy) have strengths and weaknesses. An extraversioned reference offers no direct actionable access, unlike the next two kinds. On the other hand, an intraversioned reference only discloses the version number via fragile convention ("V4"), and an introversioned reference has the pros and cons of opaque identification (e.g., hiding version numbers may be good for longevity but bad for inferring provenance).

Common content reference points

Benefits of all three forms might be realized by issuing introversioned references and offering standard reference points (elements) to common pieces of version and change information. Each of the following terms is meant to accompany an id string for specific content:

- *history*: a human-readable document that either describes the change history of the object, lists all the versions (including prior and subsequent versions), or both.
- *stabler*: fundamentally equivalent content (as permitted by the provider's content variance policy) that is expected to be available for a longer period of time, e.g., the identifier leads to an Ubuntu operating system "alpha" release but the user wants a "long term stable (LTS)" link valid for the next five years.
- *bestest*: fundamentally equivalent content (as permitted by the provider's content variance policy) that is expected, at the time of access, to be at its peak, e.g., policy permitting, with the latest and greatest features; sometimes known as "best available version".

Whatever content reference (id string) one starts with, it may not be the one that you wish to cite. For long-term citation, the given id string might or might not be suited, and rather than pausing to evaluate it by inspection, you could simply request a *stabler* (the first term above) version of the content. This is important for providers that support longer-term access to only some of their content versions, which is likely a more affordable proposition than supporting on-demand snapshotting (cf. Pröll & Rauber 2013), let alone any version at all. Conversely, for a data paper announcing your dataset, you may wish to encourage citation of the *bestest* version. The last of these terms (*history*) refers to a free-form document that is a placeholder for provenance information. There is room for a future term for a machine-readable chain or graph of content derivations; once again, if the relationships are accurately represented, it would not matter whether the content were intraversioned or extraversioned.

There are two other important content classes (**Figure 1**) that a provider could offer as common reference points, again via actionable id strings. Given any object's id string, whether for a landing page or a spreadsheet, it should be possible to use it to discover, or even to construct, id strings to related content.

- *landing*: content intended mostly for human consumption, such as an object description and links to primary information (e.g., an image file or a spreadsheet), to alternate versions and formats, and to related information; from "landing page", this is intended to support a browsing experience of an abstract overall view of the object.
- *plunging*: content intended as primary object information, often required or directly usable by software; from "below the landing page", this is intended to support an immersive object experience that bypasses any browsing step.

Desire from the scholarly community for reliable access to all information about each object has led to proposals to require that PIDs lead only to landing pages, which in theory a human being could use to recover all object information. Unfortunately, they have the unpleasant side-effect of prohibiting

mikebaird@flickr

Figure 1: Here is an informal visual aid for the *landing* vs *plunging* metaphor. A user looking for content (first image) may wish to browse among options from a spot nearby or to dive directly into already-selected content. Two vital use cases (second and third images) for persistent identifiers, the former is a *landing* experience and the latter is a *plunging* experience.

durable identifiers from leading directly to primary content. But both kinds of reference point are easily supported, especially if related content references can be *constructed* as described in the next section.

Constructed content references and identifier inflections

While reference points can be parsed out of a returned metadata record, or indeed out of hidden markup in a landing page, they can also be constructed, for example, in the manner of ARK identifier inflections (Peyrard, Kunze & Tramoni 2014). A related proposal has been made for "signposting the scholarly web" (Van de Sompel & Nelson 2015) via use of standard (Internet Assigned Numbers Authority) typed reference link relations in HTTP (Nottingham 2010). Constructed references described here have significant aspects in common with that proposal, and an alignment of terms could be useful in reducing confusion. While that proposal describes a convention for link discovery, this section describes a convention for inflecting an existing reference link to create a new link.

- *inflection*: a change to the ending of an object's id string in order to obtain a reference to content related to the originally referenced content.

Inflections are not meant for the average user. They are intended to make it easier for information specialists to explore services, create tools, and troubleshoot, similar to how internet application development became easier with the adoption of simple text-based network protocols (e.g., HTTP). The user (or agent) would start with the given id string and build a new id string by adding a query string based on one of the above terms. For example, an author might wish to include an image by direct reference from within a document, using an id string for a JPEG file such as:

http://example.org/12345/6789.jpg

Then a provider or resolver could support these "standard" constructed references:

```
http://example.org/12345/6789.jpg?goto(landing)
http://example.org/12345/6789.jpg?goto(stabler)
http://example.org/12345/6789.jpg?goto(bestest)
http://example.org/12345/6789.jpg?goto(history)
http://example.org/12345/6789.jpg?goto(plunging)
```

Effectively these are standardized query strings. Web standards have generally been loath to assign semantics to query strings, maintaining a "hands off" policy so that servers would be free to assign service-specific semantics. Because one of the proposed standard query strings above might conflict with pre-existing local semantics, we suggest that a client require server acknowledgement before assuming that it is observing the standard query string. For example, the inflection handshake for ARK identifiers breaks down if the server does not respond with an HTTP header acknowledging the THUMP protocol (Kunze & Nassar 2017).

Note that these constructed reference points do not use links from within the JPEG file, but links associated (by the proposed inflection convention) with the JPEG's id string that the provider elects to support. Construction of such references depends neither on the JPEG nor on human or machine examination of it. As a separate note, while the proposed vocabulary terms are in this case being employed in the user's request, they could in other situations be returned as element labels in metadata returned by the provider. Finally, the last of the above references (*plunging*, in this case) may in fact refer to content identical to the original id string (i.e., the image file).

With inflections it would be easy, for example, to support both *landing* and *plunging* experiences. A user or agent could thus easily request a landing page experience from any "community" id string (whose providers honor these inflections) found in the wild. There no reason that a landing page could not also contain typed links meant to be navigable by software.

To be widely useful, community conventions on the syntax of such query strings would need to be established. Internet standards makers have traditionally hesitated to specify such conventions in order not to interfere with providers' ability to make free use of URL query strings. To reduce harmful side effects, it makes sense to restrict these conventions to the community using so-called persistent identifiers (ARKs, DOIs, Handles, PURLs, URNs), and to stipulate positive acknowledgement when a provider is responding to a community standard request rather than applying a legacy interpretation (this is easily done with one extra HTTP response header), as described by Kunze & Nassar (2017). That particular method also uses simple URL modification to make it easy to request machine-readable metadata without requiring (or precluding) the more complex method of traditional HTTP content-negotiation (which requires both a URL and a separate request header). For example,

http://example.org/98765.json?show(brief)as(json)

requests a brief metadata record in the JSON format without requiring either (a) the original content to *not* be in the JSON format or (b) a tool to tweak the HTTP request headers.

Remediation

What action will be taken if there is a problem (e.g., missing content), and at what priority? Realistically, not all objects are equally important to a provider and its audience. Better supporting some objects means lowering support priority for others.

- *high*: The object receives this provider's highest priority.
- *standard*: The object receives less than this provider's highest priority.

The provider may have assigned the object identifier according to certain naming practices.

- *non-reassignment* (NR): Once assigned and made public, the identifier will not be reassigned.
- *opaque identifiers* (OP): The base identifier is assigned with no externally recognizable semantic information.
- *check character* (CC): A check character is incorporated in the assigned identifier to guard against common transcription errors.

The nature of the provider

Anyone can promise anything, but we might value a promise from one source more than from another. In some disciplines identified objects are hosted at more than one source for reasons that may include preservation, high-availability, regional language support, or added value for locally or nationally funded content enrichment. Examples include MODIS satellite imagery, biodiversity specimen data, and PubMed citations for biomedical literature.

How to characterize a given provider? Besides organizational name, identifier, and contact information, relevant factors include the provider's mission, profit motive, and succession plan. Two major provider categories are *repository* (content storage and archiving) and *resolver* (identifier service and forwarding).

- *name*: Full name of the provider organization.
- *identifier*: Unique identifier for the organization.
- *business model*: For profit (FP) or not for profit (NP).
- *mission*: One sentence mission statement of the organization.

A crucial part of persistence is sustainability of the provider organization. The National Oceanic and Atmospheric Administration (NOAA) is one of a number of government agencies around the world archiving climate data records (CDRs), which are intended to support "measurements of sufficient length, consistency, and continuity to determine climate variability and change" (Yang et al. 2016). By implication, CDRs going back many decades are intended to be made available for many decades in the future, however, this commitment must be counterbalanced, in NOAA's case, by its dependency on annual approval of national-level funding. In the case of figshare, policy is that deposited data will persist for a minimum of 10 years, and it has implemented an organizational succession plan through CLOCKSS (figshare 2016). The relatively new Open Science Framework has established an endowment fund to continue hosting data for a limited period of time in case it ceases operations (OSF 2016). Notably, to the extent that these repository behaviors and policies are revealed at all, they are not being made available as part of any object-level metadata or service.

- *succession*: The plan for dealing with sudden loss of provider viability, including set-aside funding and length of time that operations would be able to support continued operation while a successor provider is found to keep references intact.
- *certification*: If certified, acronym for certification organization or standard (e.g., TRAC, TDR, DSA) and year of certification.

Although information about the capacity of a provider resembles a trusted repository audit (Yakel et al. 2013) in miniature, the above elements are intended to capture only self-declared abilities. What this approach lacks in terms of external assessment and perception we expect to be mitigated by cost savings and emerging reputation mechanisms. Over time we hope to learn more about the utility and trustworthiness of providers and persistence statements.

Review and testing

No shortage of testing subjects is anticipated. The authors expect to pilot persistence statements within the EZID.cdlib.org system, which manages dataset identifiers for 150 paying users on three continents, from domains spanning biomedicine, earth science, and the humanities.

In an idealized scenario, the set of terms we have developed could be used in several ways. A provider could use it as a conceptual framework to perform an analysis and prioritization of its own service definition, creating policies and assigning them to various classes of objects that it stewards. While simply documenting these policies as public documentation would be useful, the provider could go further and translate the policies into metadata assertions using these terms, namely, into machine-readable persistence statements. In response to a query about the "stickiness" of a given identifier's associated object, the provider could look up the object class and return the statement it had defined for that class. The provider would also want to choose which metadata syntaxes or serializations it will support.

In a more realistic scenario, however, terms will need refining. Like most metadata creation efforts, ours ended with a well-intentioned but untested set of terms for the job at hand, namely, helping scholars gauge which objects are suitable for their citation purposes. The trick is to find the balance between metadata that producers and their tools can feasibly provide and metadata that consumers (and their tools) can feasibly interpret and find useful. Questions on both sides remain to be answered. On the producer side, can repositories populate these terms in practice? And are they willing to do so? Further, can the metadata be made available, at a technical level, via inflections or other means? Will most repositories find their persistence statements to be homogeneous across their content? If not, what kinds of heterogeneity are encountered, and how can it be handled? Finally, everything changes, including provider-side support policies, so what does it mean when persistence statements themselves change?

The consumer side most of us can relate to. Given a set of repositories providing this metadata, how can the consumer access it? Beyond that, are these terms providing the right kind of information, and sufficient information, to enable consumers to make informed decisions? Do these terms provide too much detail, calling for "consolidated ratings" (Byrnes 2000) that surface a handful of common combinations from a welter of rarely occurring combinations? And what kinds of decisions are supported in practice? For example, if a certain object is available in multiple places via multiple identifiers, can the consumer use this metadata to find the "best" citation for their purposes?

Given the burden of developing automated tools, it may be cost-effective to test persistence statements that need only be human readable until such time as confidence in the terms is established. Thus a provider could participate in testing by identifying a handful of broad object categories, writing up the associated persistence policies in a narrative that specifically references our persistence terms, and returning a reference to the appropriate narrative when any object's persistence statement is requested (e.g., with inflections). We will be seeking feedback and testers in the coming year.

Implementation

Our terms are published as YAMZ.net "vernacular" (works in progress) in order to facilitate community review and solicit feedback (to view just the persistence-related dictionary terms, please visit http://yamz.net/tag/persistence). Besides being an open platform for publishing and commenting on terms, YAMZ assigns each term a globally unique persistent identifier. As with any other persistent identifier, one should one day be able to query these terms to get persistence statements (not yet supported in YAMZ), which should reflect the malleable nature of terms in the vernacular class, contrasted with the stability expected of a separate "canonical" class. Registered users can comment, up- or down-vote a term, or "watch" a term in order to track comments and changes. Anyone can register and create new terms, and reputation-based voting prevents gaming the system. This vetting process should help our persistence terminology become more robust and useful for the communities most likely to use them.

Next steps include drafting a guidelines document to help providers and scholars use the terms. The guidelines and the terms themselves are expected to evolve through use and community feedback. As with software, early metadata adopters need to be able to tolerate small amounts of instability that come with being at the leading edge. Later adopters would enjoy a stable document with YAMZ terms that have achieved "canonical" status.

Conclusion

Persistence is a complex subject with little standardized terminology. We are accustomed to making assumptions about the long term availability of a digital object using a grab bag of explicit and implicit claims about a provider. Many providers make no promises but we nonetheless go on the assumption that an object will be available for at least the near term of our need for it. Most approaches involve best affort at long lived information systems, with less clarity about the ability to make good on implied promises.

Ours is a starter vocabulary and we are just beginning to ask for feedback. The hope is that persistence statements using the vocabulary will express qualities of digital stickiness that enable system designers to consciously reflect on the nuances of commitment and relay them honestly to users. Given the proliferation of digital repositories and collections, clarity on these matters can help users choose what to cite and help content holders choose where to deposit.

Acknowledgements

This work grew out of a breakout group discussion at a 2016 PHOIBOS2 identifier workshop in Oracle, Arizona. We are grateful to Geoff Bilder for his review comments.

Competing Interests

The authors have no competing interests to declare.

References

Abrams, S, Cruse, P, Kunze, J and **Minor, D** 2011 Curation Micro-Services: A Pipeline Metaphor for Repositories. *Journal of Digital Information*, 12(2). https://journals.tdl.org/jodi/index.php/jodi/article/view/1605/1766.

Altman, M and **King, G** 2007 A Proposed Standard for the Scholarly Citation of Quantitative Data. *D-Lib Magazine* 13(3/4). DOI: https://doi.org/10.1045/march2007-altman

Ball, A 2010 Preservation and curation in institutional repositories. Edinburgh, UK: Digital Curation Centre. Available at: http://www.dcc.ac.uk/sites/default/files/documents/reports/irpc-report-v1.3.pdf [Last accessed 12 April 2017].

Berners-Lee, T 1996 Generic Resources. Available at: https://www.w3.org/DesignIssues/Generic [Last accessed 20 May 2017].

Bilder, G 2013 DOIs unambiguously and persistently identify published, trustworthy, citable online scholarly literature. Right? Available at: https://www.crossref.org/blog/dois-unambiguously-and-persistently-identify-published-trustworthy-citable-online-scholarly-literature-right/ [Last accessed 22 May 2017].

Broeder, D and **Lannom, L** 2014 Data Type Registries: A Research Data Alliance Working Group. *D-Lib Magazine* 20(1/2). DOI: https://doi.org/10.1045/january2014-broeder

Byrnes, M 2000 Defining NLM's commitment to the permanence of electronic information. *ARL: A Bimonthly Report on Research Library Issues and Actions from ARL, CNI, and SPARC*, 212: 8–9. http://www.arl.org/storage/documents/publications/arl-br-212.pdf.

DataONE 2015 Immutability of Content in DataONE. https://releases.dataone.org/online/api-documentation-v2.0/design/ContentImmutability.html.

figshare 2016 Preservation Policies. Available at: https://support.figshare.com/support/solutions/articles/6000079077-preservation-policies [Last accessed 22 May 2017].

Hey, T, Tansley, S and **Tolle, K** 2009 The Fourth Paradigm: Data-Intensive Scientific Discovery. Redmond, Washington: Microsoft Research.

Jones, S M, Van de Sompel, H, Shankar, H, Klein, M, Tobin, R and **Grover, C** 2016 Scholarly Context Adrift: Three out of Four URI References Lead to Changed Content. *PLoS ONE*, 11(12): e0167475. DOI: https://doi.org/10.1371/journal.pone.0167475

Klein, M, Van de Sompel, H, Sanderson, R, Shankar, H, Balakireva, L, et al. 2014 Scholarly Context Not Found: One in Five Articles Suffers from Reference Rot. *PLoS ONE*, 9(12): e115253. DOI: https://doi.org/10.1371/journal.pone.0115253

Kunze, J 2002 A Metadata Kernel for Electronic Permanence. *Journal of Digital Information*, 2(2). http://journals.tdl.org/jodi/article/view/43.

Kunze, J and **Nassar, N** 2017 The HTTP URL Mapping Protocol. Work in progress. https://tools.ietf.org/html/draft-kunze-thump-03.

Kunze, J and **Rodgers, R** 2008 The ARK Identifier Scheme. *California Digital Library Publication*. https://n2t.net/ark:/13030/c7cv4br18.

Masinter, L 2012 The 'tdb' and 'duri' URI schemes, based on dated URIs. Work in progress. https://tools.ietf.org/html/draft-masinter-dated-uri-10.

Michener, W, Vieglais, D, Vision, T, Kunze, J, Cruse, P and **Janée, G** 2011 DataONE: Data Observation Network for Earth — Preserving Data and Enabling Innovation in the Biological and Environmental Sciences. *D-Lib Magazine*, 17(1/2). DOI: https://doi.org/10.1045/january2011-michener

Moats, R 1997 RFC2141: URN Syntax. https://tools.ietf.org/html/rfc2141.

Nottingham, M 2010 RFC5988: Web Linking. https://tools.ietf.org/html/rfc5988.

Nottingham, M and **Hammer-Lahav, E** 2010 RFC5785: Defining Well-Known Uniform Resource Identifiers. https://tools.ietf.org/html/rfc5785.

OSF 2016 Open Science Framework – Frequently Asked Questions. Available at: https://osf.io/faq/ [Last accessed 18 May 2017].

Peyrard, S, Kunze, J and **Tramoni, J** 2014 The ARK Identifier Scheme: Lessons Learnt at the BnF and Questions Yet Unanswered. *Proceedings of the international conference on Dublin Core and metadata applications 2014*. http://dcpapers.dublincore.org/pubs/article/view/3704/1927.

Pröll, S and **Rauber, A** 2013 Scalable data citation in dynamic, large databases: Model and reference implementation. *2013 IEEE International Conference on Big Data*. DOI: https://doi.org/10.1109/BigData.2013.6691588

Van de Sompel, H and **Nelson, M** 2015 Reminiscing About 15 Years of Interoperability Efforts. *D-Lib Magazine*, 21(11/12). DOI: https://doi.org/10.1045/november2015-vandesompel

Van de Sompel, H, Nelson, M and **Sanderson, R** 2013 RFC7089: HTTP Framework for Time-Based Access to Resource States – Memento. https://tools.ietf.org/html/rfc7089.

W3C 2014 Manual of Style. Available at: https://www.w3.org/2001/06/manual/ [Last accessed 22 May 2017].

Weigel, T, DiLauro, T and **Zastrow, T** 2014 PID Information Types: Final Report. Available at: https://doi.org/10.1045/march2007-altman [Last accessed 22 May 2017].

Yakel, E, Faniel, I, Kriesberg, A and **Yoon, A** 2013 Trust in digital repositories. *International Journal of Digital Curation*, 8(1): 143–156. DOI: https://doi.org/10.2218/ijdc.v8i1.251

Yang, W, John, V, Zhao, X, Lu, H and **Knapp, K** 2016 Satellite Climate Data Records: Development, Applications, and Societal Benefits. *Remote Sensing*, 8(4), Article 331. DOI: https://doi.org/10.3390/rs8040331

How to cite this article: Kunze, J, Calvert, S, DeBarry, J D, Hanlon, M, Janée, G and Sweat, S 2017 Persistence Statements: Describing Digital Stickiness. *Data Science Journal,* 16: 39, pp. 1–11, DOI: https://doi.org/10.5334/dsj-2017-039

Submitted: 01 November 2016 **Accepted:** 30 June 2017 **Published:** 14 August 2017

]u[*Data Science Journal* is a peer-reviewed open access journal published by Ubiquity Press.

OPEN ACCESS